Übung 3: How to build a Matrix Machine.

In dieser Übung werden wir uns eingehend mit Schleifen befassen, dabei lernen wir auch Variablen-Arrays kennen. Die Aufgabenstellung ist dabei sehr einfach:

Aus vorgegebene Kurven (figures) soll eine Matrix aus allen möglichen Kombinationen (configurations) dieser Kurven erstellt werden.

Schritt 1: Vorbereitungen

Zu Beginn sollten einige Kurven als Grundfiguren für die Matrix vorbereitet werden. Für die ordungsgemäße Funktion des Scripts sind einige Einschränkungen zu beachten:

- Zeichne eine cubic CV-Curve mit 2 Spans (5 Cvs) auf einem 4x4 Raster.
- Kopiere die Kurve und verschiebe sie um 4 Einheiten in X-Richtung.
- Bewege die Cvs, innerhalb des 4x4 Rasters.
- Kopiere erneut.

Auf diese Weise sollten insgesamt 8 Kurven erzeugt werden. Die Translate-Werte der Kurevn in ihrem 4x4 Raster soll ihrer realen Position entsprechen, wenn man TranslateX/TranslateY/TranslateZ auf Null setzt, sollte sich die Kurve wieder im ersten der 4x4 Rasterfeld befinden (deswegen das Kopieren und verschieben)

Die Aufgabe der Machine wird sein, Konfigurationen aus allen möglichen Kombinationen der Kurven zu generieren. Dabei nutzen wir die beiden großen Vorteile der Maschine: Sie ist sehr präzise (sie kann nicht anders) und sehr schnell.

Schritt 2: Erste Matrix Spalte

Kopien der ersten Originalkurve werden entlang der Y-Achse mit 4 Einheiten Abstand aufgereiht.

```
for ($i=0; $i<8; $i++)
{
	float $moveY= $i*4;
	select "curve1";
	duplicate -rr;
	move -a 0 $moveY 0 ;
}
```

Schritt 3: 2D Feld aus gleichen Kurven

Um ein zweidimensionales Feld aus Kurven zu erzeugen, packen wir diesen Code in eine weitere Schleife ein.

```
for ($i1=0; $i1<8; $i1++)
{

float $moveX=$i1*4;

	for ($i=0; $i<8; $i++)
	{
		float $moveY= $i*4;
		select "curve1";
		duplicate -rr;
		move -a $moveX $moveY 0 ;
	}

}
```

Schritt 4: 2D Feld aus unterschiedlichen Kurven

Um bei allen Durchgängen eine andere Grundkurve zu verwenden, müssen die Objekte selber Variabel sein. Deswegen speichern wir die Kurvennamen in einem string Array.

```
string $allCurves[8]={"curve1", "curve2", "curve3", "curve4", "curve5", "curve6", "curve7",
"curve8"};

for ($i1=0; $i1<8; $i1++)
{

float $moveX=$i1*4;

        for ($i=0; $i<8; $i++)
        {
                float $moveY= $i*4;
                select $allCurves[$i1];
                duplicate -rr;
                move -a $moveX $moveY 0 ;
        }

}
```

Schritt 5: 2D Matrix aus Kurvenpaaren

Für Konfigurationen werden immer zwei Kurven kombiniert. D.h. In jedem Durchgang der inneren Schleife werden jeweils zwei Kurven dupliziert und an die jeweiligen Koordinaten verschoben. Der Unterschied:
Eine der Kurvenvariablen ändert sich nur nach dem Zähler der äußeren Schleife, die andere nach dem der inneren.

```
string $allCurves[8]={"curve1", "curve2", "curve3", "curve4", "curve5", "curve6", "curve7",
"curve8"};

for ($i1=0; $i1<8; $i1++)
{

float $moveX=$i1*4;

        for ($i=0; $i<8; $i++)
        {
                float $moveY= $i*4;
                select $allCurves[$i1];
                duplicate -rr;
                move -a $moveX $moveY 0 ;

                select $allCurves[$i];
                duplicate -rr;
                move -a $moveX $moveY 0 ;
        }

}
```

Schritt 6: Dynamische Kurvenauswahl

Das Skript wird noch wesentlich flexibler, wenn die Kurven der Matrix nicht im Script vordenfiniert sind, sondern in der Maya Scene direkt ausgewählt werden können. Das bedeutet, daß sowohl die Namen, als auch die Zahl der Objekte variabel sein müssen.

```
string $allCurves[]=`ls -sl`;

int $numCRV=size($allCurves);

for ($i1=0; $i1<$numCRV; $i1++)
{

float $moveX=$i1*4;

	for ($i=0; $i<$numCRV; $i++)
	{
		float $moveY= $i*4;
		select $allCurves[$i1];
		duplicate -rr;
		move -a $moveX $moveY 0 ;

		select $allCurves[$i];
		duplicate -rr;
		move -a $moveX $moveY 0 ;
	}

}
```

Schritt 7: Kurven aneinander setzen

Bisher werden die Kurven immer nur übereinandergelegt. Im nächsten Schritt sollen die zweite Kurve jeweils ans Ende der ersten angesetzt werden. Dazu müssen die Koordinaten der Endpunkte erst abgefragt und in einem float[] array gesepeichert werden.

```
string $allCurves[]=`ls -sl`;

int $numCRV=size($allCurves);

for ($i1=0; $i1<$numCRV; $i1++)
{

float $moveX=$i1*8;

	for ($i=0; $i<$numCRV; $i++)
	{
		float $moveY= $i*8;
		select $allCurves[$i1];
		duplicate -rr;
		move -a $moveX $moveY 0 ;

		string $qEnd=($allCurves[$i1]+".cv[4]"); //string mit Namen des 5. CVs
		float $end[]=`getAttr $qEnd`; //Abfrage der Koordinaten des CVs

		select $allCurves[$i];
		duplicate -rr;

		string $qStart=($allCurves[$i]+".cv[0]"); //string mit Namen des 1. CVs
		float $start[]=`getAttr $qStart`; // Abfrage der Koordinaten des ersten Cvs.

		move -a $moveX $moveY 0 ;
		move -r ($end[0]-$start[0]) ($end[1]-$start[1]) ($end[2]-$start[2]);
	}
}
```